

Overlapping community detection on attributed graphs via neutrosophic C-means

Yuhan Jia(贾雨涵), Leyan Ouyang(欧阳乐严), Qiqi Wang(王萁淇), and Huijia Li(李慧嘉)
Citation: Chin. Phys. B, 2025, 34 (**12**): 128901. DOI: 10.1088/1674-1056/adea9a
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Extracting fuzzy clusters from massive attributed graphs using Markov lumpability optimization

Kai-Yue Jiang(蒋凯悦), Li-Heng Xu(徐力恒), Shi-Pei Lin(林诗佩), Li-Yang Zhou(周李阳), Hui-Jia Li(李慧嘉), and Ge Gao(高歌)
Chin. Phys. B, 2025, 34 (**10**): 108903. DOI: 10.1088/1674-1056/addaa2

SFFSlib: A Python library for optimizing attribute layouts from micro to macro scales in network visualization

Ke-Chao Zhang(张可超), Sheng-Yue Jiang(蒋升跃), and Jing Xiao(肖婧)
Chin. Phys. B, 2025, 34 (**5**): 058903. DOI: 10.1088/1674-1056/adcb98

Exploring the proper statistical proxy to distinguish random structures

Mei Xie(解梅), Fei Xie(谢飞), Baoyu Song(宋宝玉), Qiaoyu Guo(郭桥雨), and Xuechen Jiao(焦学琛)
Chin. Phys. B, 2025, 34 (**5**): 058902. DOI: 10.1088/1674-1056/adbd2b

Characteristics of complex network of heatwaves over China

Xuemin Shen(沈雪敏), Xiaodong Hu(胡晓东), Aixia Feng(冯爱霞), Qiguang Wang(王启光), and Changgui Gu(顾长贵)
Chin. Phys. B, 2025, 34 (**3**): 038903. DOI: 10.1088/1674-1056/ada43a

A local fuzzy method based on “p-strong” community for detecting communities in networks

Yi Shen(沈毅), Gang Ren(任刚), Yang Liu(刘洋), Jia-Li Xu(徐家丽)
Chin. Phys. B, 2016, 25 (**6**): 068901. DOI: 10.1088/1674-1056/25/6/068901

Overlapping community detection on attributed graphs via neutrosophic C-means

Yuhan Jia(贾雨涵)[†], Leyan Ouyang(欧阳乐严)[†], Qiqi Wang(王其淇)[‡], and Huijia Li(李慧嘉)[§]

School of Statistics and Data Science, AAIS, LPMC and KLMDASR, Nankai University, Tianjin 300071, *China*

(Received 3 April 2025; revised manuscript received 23 June 2025; accepted manuscript online 2 July 2025)

Detecting overlapping communities in attributed networks remains a significant challenge due to the complexity of jointly modeling topological structure and node attributes, the unknown number of communities, and the need to capture nodes with multiple memberships. To address these issues, we propose a novel framework named density peaks clustering with neutrosophic C-means. First, we construct a consensus embedding by aligning structure-based and attribute-based representations using spectral decomposition and canonical correlation analysis. Then, an improved density peaks algorithm automatically estimates the number of communities and selects initial cluster centers based on a newly designed cluster strength metric. Finally, a neutrosophic C-means algorithm refines the community assignments, modeling uncertainty and overlap explicitly. Experimental results on synthetic and real-world networks demonstrate that the proposed method achieves superior performance in terms of detection accuracy, stability, and its ability to identify overlapping structures.

Keywords: attributed graphs, overlapping communities, neutrosophic C-means, density peaks

PACS: 89.75.Fb, 89.75.–k

DOI: 10.1088/1674-1056/adea9a **CSTR:** 32038.14.CPB.adea9a

1. Introduction

Community detection is a fundamental problem in complex network analysis, aiming to identify groups of nodes that are densely connected within a network.[1,2] It plays a critical role in understanding the structural and functional properties of complex networks.[3–5] Many real-world network systems comprise numerous highly interconnected entities, each characterized by diverse attribute information.[6] These systems are often effectively modeled as attributed graphs, where nodes represent users, attributes capture user profiles, and edges denote interactions. A fundamental challenge is to partition attributed graphs into meaningful clusters that simultaneously preserve topological cohesion and attribute similarity, a process known as attributed graph clustering (AGC).[7,8] AGC has a broad range of applications across diverse domains, including the Internet, protein–protein interaction networks, trade networks, and scientific collaborations.[9–12]

Moreover, it has been shown that the very existence of overlapping communities is one of the most important characteristics of various complex networks where some nodes may connect and belong to multiple communities and play a critical role in the flow of information. However, traditional methods struggle to effectively integrate these two types of information and frequently fail to handle the overlapping communities commonly observed in real networks. Specifically, these approaches face two key challenges:

Lack of a simple and efficient integration of attributes and topology Most algorithms focus exclusively on either network topology (e.g., modularity optimization[13]) or node attributes (e.g., attribute clustering), neglecting their inherent correlations. For example, random walk-based methods such as Node2Vec,[14] capture topological similarities but remain insensitive to attribute information, while matrix factorization approaches such as AGNMF-AN[15] prioritize attribute homogeneity at the expense of structural patterns. This artificial separation leads to biased and incomplete community representations. Although recent studies have attempted to bridge topology and attribute information through unified models, this remains a challenging and active area of research. Methods based on joint loss optimization,[16] attribute semantics and structural learning,[17] and matrix co-factorization[18] have achieved promising results. However, these approaches often require complex model design, heavy parameter tuning, and may lack interpretability. Therefore, there is still a need for a solution that offers a simple and efficient integration of both aspects, providing a lightweight, unsupervised, and explainable fusion mechanism.

Result instability due to manual parameter specification Many advanced methods, including fuzzy C-means (FCM) variants and neutrosophic C-means (NCM),[19] require predefining the number of communities, i.e., a parameter often unknown in real-world applications. This manual specification introduces subjectivity and causes significant instability, as different initializations may yield divergent community structures. While density-based algorithms such as density

[†]These authors contributed equally to this work.

[‡]Corresponding author. E-mail: qiqi.wang@nankai.edu.cn

[§]Corresponding author. E-mail: hjli@nankai.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

peaks[20] can detect cluster counts, they have not been adapted for attributed networks, leaving a critical gap in automated community detection. Recent learning-based approaches, including reinforcement[21] and transfer learning,[22] aim to reduce manual tuning by predicting hyperparameters from network features.[23] Nonetheless, these approaches often require substantial computational resources, labeled data, and pretraining, and they may sacrifice interpretability and automation. We aim to improve the parameter instability issue while preserving the advantages of FCM-like methods, such as their ability to handle fuzzy and overlapping community structures.

To address these problems, we propose a new framework, named density peaks clustering with neutrosophic C-means (DPCNCM), which integrates enhanced density peak (DP) clustering with NCM to achieve robust overlapping community detection on attributed graphs. This approach is conceptually simple and avoids complex training. It relies on spectral embedding and canonical correlation analysis (CCA), which are easy to implement and require no backpropagation or parameter tuning compared with deep learning-based methods. Subsection 5.3 demonstrates the low computational complexity of DPCNCM, and its convergence behavior, reported in Subsection 6.4, further confirms that the algorithm does not have a complex training process. Moreover, the visualizations of overlapping communities in Subsection 6.4 illustrate that DPCNCM is capable of revealing clear and meaningful structures in real-world networks.

Our key contributions include the following. (i) We introduce the neutrosophic theory for community detection in an attributed graph, which enables the calculation of membership degree to certain/uncertain communities and the unassigned point set. Specifically, a modified loss function incorporating true/indeterminate/false membership values with Lagrange constraints, explicitly modeling uncertainty and tendency to remain unassigned to ensure consistent optimization. (ii) An improved automated DP algorithm is introduced by designing a cluster strength (CS) metric that quantitatively evaluates cluster centers based on local density and separation. By combining with segmented least squares regression, we can automatically determine the optimal number of communities and their centers, which significantly enhances stability.[20,24,25] (iii) The enhanced NCM component leverages DP-derived centers for initialization, replacing random initialization with density-guided distance matrices. We can substantially improve clustering stability while effectively combining network topology and node attributes.

Finally, experiments on both synthetic and real-world networks demonstrate that DPCNCM outperforms other algorithms in community detection accuracy and identifying overlapping structures. On synthetic networks with varying mixing parameters, DPCNCM achieves superior results in terms of ONMI and Omega, effectively capturing overlapping communities. Additionally, it exhibits robust performance across networks of different sizes. In real-world networks, DPCNCM accurately identifies a reasonable number of communities, uncovers meaningful overlapping structures, and achieves high modularity.

2. Related works

2.1. Attributed network embedding

In real-world networks, nodes are often not only connected through topological structures but also accompanied by abundant attribute information. For instance, in social networks, user behaviors, interests, and social relationships can serve as node attributes, and these attributes exhibit strong correlations with the network structure.[26] Previous methods for encoding attributed networks can be broadly categorized into two types: random walk-based methods and matrix factorization-based methods. For the first type, the mathematical theory of random walks was first proposed by Pearson *et al.* in 1905[27] and has since been widely used to measure node proximity in network topologies. Yang *et al.* proposed the SA-cluster algorithm, which introduces a unified distance metric to effectively combine structural and attribute similarities, ensuring both structural compactness and attribute homogeneity during clustering.[28] Some studies model node attributes as bipartite graphs, which increases the diversity of random walks by incorporating attribute edges and enables the use of deep-learning models to optimize embedding representations.[29–31] However, these algorithms rely on random walk-based distance calculations, which are computationally expensive and susceptible to noise and high-dimensional sparsity.[14,31,32] For the second type, matrix factorization methods are conceptually clear and computationally efficient. For example, Kamal Berahmand *et al.* proposed the AGNMF-AN method, which constructs an attribute-augmented graph to integrate structural and attribute information and introduces an adaptive similarity matrix with an $L_{2,1}$-norm to improve community detection performance.[15] Additionally, spectral theory and CCA-based network embedding methods map topological structure and node attribute information into a low-dimensional Euclidean space, reducing noise in the data and generating a consensus embedding matrix through CCA.[26,33] This approach is conceptually simpler, avoids complex training processes, and offers lower computational complexity and stronger interpretability. In this study, we adopt this method as the network embedding approach for attributed graphs and further extend it to overlapping community detection.

2.2. Overlapping community detection

Overlapping community detection methods can be broadly categorized into membership propagation-based

methods and probabilistic models, each employing distinct strategies to identify communities with overlapping nodes. For the first category, membership propagation-based methods determine community affiliations by iteratively propagating membership information across the network. A prominent example is the label propagation approach, where Gregory *et al.* proposed an algorithm that incorporates node importance to improve overlapping community detection.[34] Representative methods in this category also include the community overlap propagation algorithm (COPRA), which generalizes label propagation to support multiple community memberships, and the democratic estimate of modular organization (DEMON), which detects overlaps through local expansion and voting mechanisms.[35] Another notable approach is the clique percolation method (CPM), which detects overlapping communities by identifying interconnected cliques, though its computational cost can be prohibitive for large networks.[36] On the other hand, probabilistic models employ statistical techniques to infer overlapping community structures.[37] For instance, Guo *et al.* introduced the NCM algorithm, which leverages fuzzy clustering and neutrosophic sets to model node memberships in communities, uncertain regions, and noise.[19] Similarly, Li *et al.* developed a multi-scale optimization method that refines community assignments by iteratively updating membership vectors based on topological entropy.[38] These approaches excel in handling uncertainty and noisy data, making them suitable for complex real-world networks. While propagation-based methods often prioritize scalability, probabilistic approaches provide more nuanced representations of overlapping structures. Recent advances continue to bridge these strategies, combining their strengths for more robust community detection.

3. Heterogeneous data embedding

In attributed graphs, topological structure and node attributes represent distinct and heterogeneous aspects of the data. Moreover, the data is often noisy and incomplete, making it challenging to remove noise without compromising structural or attribute integrity. To address this problem, we introduce a network embedding model that maps topological and node attribute information into Euclidean space, drawing on the work of Belkin and Niyogi on Laplacian eigenmaps for dimensionality reduction and data representation.[33] The method first applies spectral theory to project the topological and attribute information separately into Euclidean space, which helps reduce noise while preserving meaningful structure. To ensure maximum preservation of structural and attribute characteristics while integrating the denoised information, CCA is employed.[26] By maximizing the correlation between these two representations, the method generates a consensus embedding matrix that captures the essential properties of the original graph. The process is illustrated in the first stage of the overall workflow shown in Fig. 1.

3.1. Topological information embedding

Consider a heterogeneous attributed graph $\boldsymbol{G} = (\boldsymbol{U}, \boldsymbol{E}, \boldsymbol{S})$, where $\boldsymbol{U}$ denotes the set of nodes, $\boldsymbol{E}$ the set of edges, and $\boldsymbol{S}$ the node attribute matrix. Let $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ be the adjacency matrix of the graph. The corresponding degree matrix is denoted by $\boldsymbol{D}_A$, and the unnormalized graph Laplacian is defined as $\boldsymbol{L}_A = \boldsymbol{D}_A - \boldsymbol{A}$. Based on spectral theory, we solve the following eigenvalue problem:

$$\boldsymbol{L}_A \boldsymbol{a} = \lambda \boldsymbol{D}_A \boldsymbol{a}. \tag{1}$$

Selecting the first t eigenvectors, we form the following embedding matrix for the topological information:

$$\boldsymbol{X}_A = [\boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_{t+1}] \in \mathbb{R}^{N\times t}. \tag{2}$$

3.2. Attribute information embedding

For the node attribute matrix $\boldsymbol{S} \in \mathbb{R}^{N\times m}$, we first standardize the attributes to obtain the normalized matrix $\tilde{\boldsymbol{S}}$. Next, we construct the cosine similarity matrix $\boldsymbol{W}$, where the element W_{ij} represents the cosine similarity between nodes i and j:

$$W_{ij} = \frac{\tilde{\boldsymbol{S}}_i \cdot \tilde{\boldsymbol{S}}_j}{\|\tilde{\boldsymbol{S}}_i\|_2 \|\tilde{\boldsymbol{S}}_j\|_2} = \frac{\sum_{k=1}^{m} \tilde{S}_{ik}\tilde{S}_{jk}}{\sqrt{\sum_{k=1}^{m} \tilde{S}_{ik}^2}\sqrt{\sum_{k=1}^{m} \tilde{S}_{jk}^2}}. \tag{3}$$

We solve the eigenvalue problem for $\boldsymbol{W}$:

$$\boldsymbol{W}\boldsymbol{b} = \lambda \boldsymbol{b}. \tag{4}$$

The top k eigenvectors are selected to form $\boldsymbol{X}_S$, which captures the similarity of node attributes in Euclidean space.

$$\boldsymbol{X}_S = [\boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{t+1}] \in \mathbb{R}^{N\times t}. \tag{5}$$

3.3. Dual-information fused embedding

To effectively integrate the two independently obtained embedding matrices $\boldsymbol{X}_A$ and $\boldsymbol{X}_S$, CCA is employed to identify the projection matrix that maximize the correlation between the two representations after transformation. Accordingly, the consensus embedding matrix $\boldsymbol{X}$ is derived as

$$\boldsymbol{X} = [\boldsymbol{X}_A, \boldsymbol{X}_S] \cdot \boldsymbol{P} \in \mathbb{R}^{N\times l}. \tag{6}$$

Here, $\boldsymbol{P} = [\boldsymbol{p}_A; \boldsymbol{p}_S]$ is the projection matrix obtained from solving the following optimization problem:

$$\begin{aligned} \max_{\boldsymbol{p}_A, \boldsymbol{p}_S} \quad & \boldsymbol{p}_A' \boldsymbol{X}_A' \boldsymbol{X}_A \boldsymbol{p}_A + \boldsymbol{p}_A' \boldsymbol{X}_A' \boldsymbol{X}_S \boldsymbol{p}_S \\ & + \boldsymbol{p}_S' \boldsymbol{X}_S' \boldsymbol{X}_A \boldsymbol{p}_A + \boldsymbol{p}_S' \boldsymbol{X}_S' \boldsymbol{X}_S \boldsymbol{p}_S \\ \text{s.t.} \quad & \boldsymbol{p}_A' \boldsymbol{X}_A' \boldsymbol{X}_A \boldsymbol{p}_A + \boldsymbol{p}_S' \boldsymbol{X}_S' \boldsymbol{X}_S \boldsymbol{p}_S = 1. \end{aligned} \tag{7}$$

To solve the optimization problem, we construct the Lagrange equation and set its partial derivatives to zero to find

the optimal solution for $[\boldsymbol{p}_A;\boldsymbol{p}_S]$. This is equivalent to solving the following eigenvalue problem:

$$\begin{bmatrix} \boldsymbol{X}_A'\boldsymbol{X}_A & \boldsymbol{X}_S'\boldsymbol{X}_A \\ \boldsymbol{X}_A'\boldsymbol{X}_S & \boldsymbol{X}_S'\boldsymbol{X}_S \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_A \\ \boldsymbol{p}_S \end{bmatrix} = \gamma \begin{bmatrix} \boldsymbol{X}_A'\boldsymbol{X}_A & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_S'\boldsymbol{X}_S \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_A \\ \boldsymbol{p}_S \end{bmatrix}, \quad (8)$$

where $\boldsymbol{X}_A'$ is the transpose of $\boldsymbol{X}_A$, and γ is the Lagrange multiplier. $\boldsymbol{P}$ is the top-l eigenvectors. The resulting consensus embedding matrix $\boldsymbol{X}$ effectively captures the relationships between topological structure and attribute information, which can be naturally applied in clustering.

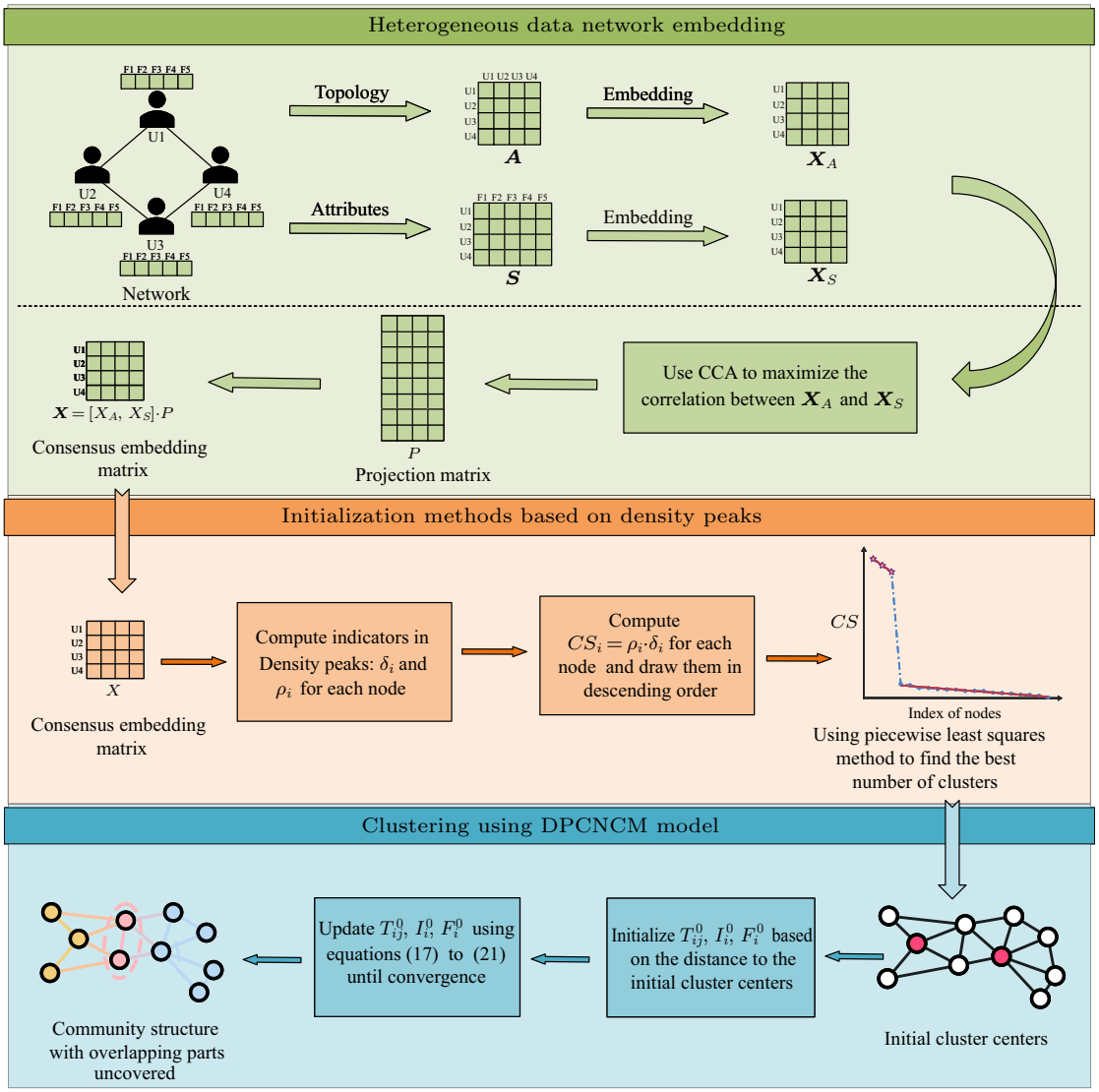


Fig. 1. The flow chart of the proposed method. It contains three main steps, i.e., 1. heterogeneous data network embedding, 2. initialization methods based on DPs, and 3. clustering using the DPCNCM method.

4. Neutrosophic community detection in the attributed graph

4.1. Neutrosophic set

Neutrosophy, proposed by Florentin Smarandache in 1995, is a philosophical theory that studies neutrality and its relationship with opposing concepts.[39] It suggests that every concept coexists with its opposite and a neutral region, making boundaries between ideas more flexible and comprehensive. Based on this theory, the neutrosophic set (NS) extends the concepts of fuzzy sets and intuitionistic fuzzy sets. For a given element x in a set M, its membership is represented as a triplet (T,I,F) where T (truth) denotes the degree to which x belongs to M, I (indeterminacy) represents uncertainty, and F (falsehood) indicates the degree to which x does not belong to M. This framework allows for a more flexible representation of uncertainty, particularly in cases with ambiguous or overlapping data.

In clustering analysis, some data points fall near the boundaries between clusters, making their assignment ambiguous. Forcefully allocating such points to a specific cluster can lead to errors in determining cluster centers. To address this issue, Guo *et al.* proposed an improved FCM[40] clustering algorithm that combines NSs and FCM.[19] Inspired by NSs, this method simultaneously considers the membership degrees of nodes to both certain and uncertain sets, viewing

all nodes as the union of certain clusters, boundary point sets, and an unassigned set. The unassigned set contains nodes that are too distant from all cluster centers to be meaningfully assigned to either determinate clusters or boundary regions. The combination enhances the algorithm's ability to handle uncertain data, which is particularly useful for detecting overlapping nodes in community detection problems.

4.2. Neutrosophic clustering method for overlapping communities

Compared to traditional FCM clustering, the NCM clustering algorithm can calculate not only the membership degree of each node to the certain cluster (T) but also the membership degrees of nodes to the uncertain clusters (I) and the unassigned set (F). The main innovation of this method is the incorporation of true/indeterminate/false membership values into the traditional FCM objective function, enabling the model to effectively quantify each node's certainty, ambiguity, and tendency to remain unassigned. In this article, we perform NCM clustering on the embedding matrix X obtained in the attributed network to capture overlapping communities, thereby enhancing the identification of community structures within networks.

The modified objective function based on FCM is defined as follows:

$$\begin{aligned}\min \quad J(\boldsymbol{T},\boldsymbol{I},\boldsymbol{F},\boldsymbol{V}) &= \sum_{i=1}^{N}\sum_{j=1}^{C}(\omega_1 T_{ij})^m \|\boldsymbol{x}_i-\boldsymbol{v}_j\|_2 \\ &+\sum_{i=1}^{N}(\omega_2 I_i)^m \|\boldsymbol{x}_i-\overline{\boldsymbol{v}}_{i\max}\|_2 \\ &+\sum_{i=1}^{N}\delta^2(\omega_3 F_i)^m, \end{aligned} \tag{9}$$

where

$$cp_i = \arg\max_{j=1,2,\ldots,C}(T_{ij}), \tag{10}$$

$$cq_i = \arg\max_{j\neq cp_i, j=1,2,\ldots,C}(T_{ij}), \tag{11}$$

$$\bar{\boldsymbol{v}}_{i\max} = \frac{\boldsymbol{v}_{cp_i}+\boldsymbol{v}_{cq_i}}{2}. \tag{12}$$

Here, m is a constant, δ is used to control the number of unassigned nodes, and ω_i represents the weight factors. cp_i and cq_i are the indices of the communities corresponding to the maximum and second maximum values of T_{ij}, respectively. $\bar{\boldsymbol{v}}_{i\max}$ is the average of the two communities $\boldsymbol{v}_{cp_i}$ and $\boldsymbol{v}_{cq_i}$. T_{ij} denotes the membership degree of node i to the certain community j, while I_i and F_i represent the membership degrees of the overlapping region and the unassigned set, respectively. C denotes the number of communities, which is usually prespecified in general methods. However, since the ground-truth number of communities is unknown, the algorithm proposed in Section 5 automatically determines an appropriate number of communities. The constraints are as follows:

$$0 < T_{ij}, I_i, F_i < 1, \tag{13}$$

$$\sum_{j=1}^{C} T_{ij} + I_i + F_i = 1. \tag{14}$$

To solve this constrained optimization problem, we can construct a Lagrange function

$$\begin{aligned} L(\boldsymbol{T},\boldsymbol{I},\boldsymbol{F},\boldsymbol{V},\boldsymbol{\lambda}) &= \sum_{i=1}^{N}\sum_{j=1}^{C}(\omega_1 T_{ij})^m \|\boldsymbol{x}_i-\boldsymbol{v}_j\|_2 \\ &+\sum_{i=1}^{N}(\omega_2 I_i)^m \|\boldsymbol{x}_i-\overline{\boldsymbol{v}}_{\max}\|_2 \\ &+\sum_{i=1}^{N}\delta^2(\omega_3 F_i)^m \\ &+\sum_{i=1}^{N}\lambda_i\left(\sum_{j=1}^{C}T_{ij}+I_i+F_i-1\right). \end{aligned} \tag{15}$$

We set the gradient of $L(\boldsymbol{T},\boldsymbol{I},\boldsymbol{F},\boldsymbol{V},\boldsymbol{\lambda})$ to zero, resulting in the following system of equations:

$$\begin{cases} \frac{\partial L}{\partial T_{ij}} = 0, \\ \frac{\partial L}{\partial I_i} = 0, \\ \frac{\partial L}{\partial F_i} = 0, \\ \frac{\partial L}{\partial \boldsymbol{v}_j} = 0, \\ \sum_{j=1}^{C} T_{ij} + I_i + F_i = 1. \end{cases} \tag{16}$$

By solving this system, we obtain

$$\boldsymbol{v}_j = \frac{\sum_{i=1}^{N}(\omega_1 T_{ij})^m \boldsymbol{x}_i}{\sum_{i=1}^{N}(\omega_1 T_{ij})^m}, \tag{17}$$

$$T_{ij} = \frac{K}{\omega_1}\left\|\boldsymbol{x}_i-\boldsymbol{v}_j\right\|_2^{-\frac{2}{m-1}}, \tag{18}$$

$$I_i = \frac{K}{\omega_2}\left\|\boldsymbol{x}_i-\bar{\boldsymbol{v}}_{i\max}\right\|_2^{-\frac{2}{m-1}}, \tag{19}$$

$$F_i = \frac{K}{\omega_3}\delta^{-\frac{2}{m-1}}, \tag{20}$$

where

$$\begin{aligned}\frac{1}{K} &= \frac{1}{\omega_1}\sum_{j=1}^{C}\left\|\boldsymbol{x}_i-\boldsymbol{v}_j\right\|_2^{-\frac{2}{(m-1)}} \\ &+\frac{1}{\omega_2}\left\|\boldsymbol{x}_i-\bar{\boldsymbol{v}}_{i\max}\right\|_2^{-\frac{2}{(m-1)}}+\frac{1}{\omega_3}\delta^{-\frac{2}{(m-1)}}. \end{aligned} \tag{21}$$

The T_{ij}, I_i, F_i and the community center $\boldsymbol{v}_j$ are updated using Eqs. (17)–(21). After random initialization of $\boldsymbol{T}$, $\boldsymbol{I}$, $\boldsymbol{F}$ under the given constraints, the clustering process is conducted through the iterative optimization of the objective function until $\left|\boldsymbol{T}^{k+1}-\boldsymbol{T}^k\right| < \varepsilon$ is achieved, where k denotes the iteration step, and ε is a pre-set parameter. Ultimately, each node is assigned to the community with the maximum $\boldsymbol{TM} = [\boldsymbol{T},\boldsymbol{I},\boldsymbol{F}]$ value.

5. Improved neutrosophic C-means method

Based on the above discussion, NCM methods suffer from instability due to random initialization and require a predefined number of communities as input. However, in real-world scenarios, the actual number of communities is typically unknown in advance (discussed in Subsection 4.2). To overcome these limitations, we propose DPCNCM, a unified framework that including: (i) an automated DP variant using quantitative cluster potential metrics to objectively determine community structures (Subsection 5.1), and (ii) an enhanced NCM implementation that leverages density-guided initialization and balanced attribute-topology integration (Subsection 5.2).

5.1. Automatic density peaks method

Among clustering algorithms capable of determining the central nodes of clusters,[41–43] the DP method has been widely used due to its speed and accuracy,[20] including applications in network community detection. The core idea of the DP clustering method is that cluster centers should have higher density than their neighboring points and be farther away from points with even higher density. Based on the definition of the center of the cluster, we present the following definitions.

Definition 1 The local density ρ_i of point i is defined as the number of points whose distance from point i is less than d_c:

$$\rho_i = \sum_j \chi\left(d_{ij} - d_c\right). \tag{22}$$

If $x < 0$, then $\chi(x) = 1$, otherwise $\chi(x) = 0$, d_c is the cutoff distance.

Definition 2 The minimum distance δ_i is defined as the minimum distance between point i and any other point with higher density:

$$\delta_i = \min_{j:\rho_j > \rho_i} \left(d_{ij}\right). \tag{23}$$

For the points with the highest density, we usually take $\delta_i = \max_j(d_{ij})$.

After computing ρ_i and δ_i for each point, a decision graph is constructed with ρ on the horizontal axis and δ on the vertical axis. The cluster center is selected as the point clearly arranged in the upper right corner in the decision graph. However, this method depends on manual inspection of decision graphs, which limits its accuracy and makes it difficult to reliably distinguish cluster centers from other points.[20] An automatic DP method using least-squares can overcome the limitations of manual observation,[44] providing precise cluster numbers and centroids. Therefore, it can be used to identify the number of communities and initial community centers in community detection.

The automatic DP method determines the optimal number of clusters and cluster centers using segmented least squares instead of relying on observing decision diagrams. First, the CS value is introduced to quantify the potential of a sample point serving as a cluster center. The CS values are sorted in descending order and plotted as a line chart, where the optimal number of clusters corresponds to the inflection point of the curve. This approach leverages the least-squares method to accurately determine the inflection point.

To provide a smooth and continuous measure of local density, we define ρ_i using a Gaussian kernel:

$$\rho_i = \sum_{j=1}^{n-1} \mathrm{e}^{\frac{d_{i,j}}{d_c}}, \tag{24}$$

where $d_{i,j}$ represents the distance between node i and node j, while d_c is the cutoff distance.

Definition 3 CS_i represents the potential of the next cluster center. In our hypothesis, the cluster center v_i should have higher ρ_i and δ_i. We define

$$CS_i = \rho_i \times \delta_i. \tag{25}$$

In this method, the decision graph plots CS_i values in descending order against the number of clusters k. As k increases, CS_i decreases monotonically, but the rate of decrease often becomes much slower starting from a certain k. We assume that cluster centers retain high CS_i values, while CS_i values for other points drop sharply. The optimal number of clusters is determined at the "knee" of the curve, identified as the point of maximum curvature. To locate this knee precisely, we model the curve as two approximately linear segments on either side of the inflection point. Using the least squares method, we fit linear models to these segments and define the optimal number of clusters as their intersection.

For a given point (k_i, CS_i), we assume a linear equation

$$CS_i = \beta_0 + \beta_1 k_i, \tag{26}$$

where β_0 and β_1 are estimated parameters, and k_i represents the serial number of the node from 1 to N.

The least-squares method minimizes the sum of squared errors:

$$\begin{aligned} S(\beta_0, \beta_1) &= \sum_i \left(\widehat{CS}_i - CS_i\right)^2 \\ &= \sum_i (\beta_0 + \beta_1 k_i - CS_i)^2. \end{aligned} \tag{27}$$

By setting the gradient of the equation equal to zero, we obtain

$$\hat{\beta}_0 = \frac{\sum(CS_i)}{n} - \hat{\beta}_1 \frac{\sum(k_i)}{n}, \tag{28}$$

$$\hat{\beta}_1 = \frac{n\sum(k_i CS_i) - \sum k_i \sum CS_i}{n\sum\left(k_i^2\right) - \left(\sum k_i\right)^2}. \tag{29}$$

Let L_c and R_c represent data points on the left and right of the optimal clustering number c, where L_c contains $k = 2, \ldots, c$

and R_c contains $k = c+1, \ldots, N$ for $c = 3, \ldots, N-2$. Using the above equations, we obtain two linear models

$$CS_{il} = \beta_{0r} + \beta_{1r} k_i, \quad i = 2, \ldots, c, \tag{30}$$

$$CS_{ir} = \beta_{0l} + \beta_{1l} k_i, \quad i = c+1, \ldots, N. \tag{31}$$

The overall root-mean-square error is then defined as follows (here L_c and R_c are divided by $x = c$):

$$\mathrm{RMSE}_c = \frac{c-1}{N-1} \times \mathrm{RMSE}(L_c) + \frac{N-c}{N-1} \times \mathrm{RMSE}(R_c), \tag{32}$$

where $\mathrm{RMSE}(L_c)$ is the root-mean-square error for the linear model that best fits the points in L_c (and similarly for L_r):

$$\mathrm{RMSE}(L_c) = \frac{\sqrt{\sum_{i=2}^{c} \left(\widehat{CS}_i - CS_i\right)^2}}{c}. \tag{33}$$

Finally, we find $\hat{c}$ by minimizing RMSE_c:

$$C = \arg\min_c \mathrm{RMSE}_c, \quad c = 2, \ldots, N. \tag{34}$$

Once the optimal number of clusters is identified, all points are sorted in descending order of their CS values, and the top-ranked points are selected as initial cluster centers.

5.2. The improvement of neutrosophic C-means

Unlike the random initialization in neutrosophic C-means, the TM values are initialized using the initial cluster centers which are denoted as $\{\boldsymbol{v}_1{}^0, \boldsymbol{v}_2{}^0, \ldots, \boldsymbol{v}_C{}^0\}$ obtained in Subsection 5.1. For nodes that do not belong to $\{\boldsymbol{v}_1{}^0, \boldsymbol{v}_2{}^0, \ldots, \boldsymbol{v}_C{}^0\}$, a smaller distance to a cluster center indicates a higher membership degree ($\boldsymbol{T}$) for that cluster. Additionally, a smaller distance to the midpoint of the two centers with the highest membership values implies a higher indeterminacy ($\boldsymbol{I}$). Then, the values of $\boldsymbol{T}$ and $\boldsymbol{I}$ are normalized to the range $[0, 1]$, while $\boldsymbol{F}$ is set to 0.5. The corresponding formulas are defined as follows:

$$T_{ij}{}^* = \frac{1}{\left\|\boldsymbol{x}_i - \boldsymbol{v}_j\right\|_2}, \quad \boldsymbol{x}_i \notin \{\boldsymbol{v}_1^0, \boldsymbol{v}_2^0, \ldots, \boldsymbol{v}_C^0\}, \tag{35}$$

$$I_i{}^* = \frac{1}{\left\|\boldsymbol{x}_i - \bar{\boldsymbol{v}}_{i\max}\right\|_2}, \quad \boldsymbol{x}_i \notin \{\boldsymbol{v}_1^0, \boldsymbol{v}_2^0, \ldots, \boldsymbol{v}_C^0\}, \tag{36}$$

$$T_{ij}^0 = \frac{T_{ij}{}^*}{\sum_{j=1}^{C} T_{ij} + I_i}, \tag{37}$$

$$I_i{}^0 = \frac{I_i{}^*}{\sum_{j=1}^{C} T_{ij} + I_i}. \tag{38}$$

For nodes belonging to $\{\boldsymbol{v}_1^0, \boldsymbol{v}_2^0, \ldots, \boldsymbol{v}_C^0\}$, if node $\boldsymbol{u}_i$ is the jth cluster center, then $T_{ij}^0 = 1$, while $T_{ik}^0 (k \neq j)$, $I_i{}^0$, $F_i{}^0$ are set to 0.

5.3. Complexity analysis

Suppose a network comprises n nodes, each with a d-dimensional attribute vector. Let A be the adjacency matrix, C the number of cluster centers, and T the number of iterations required for convergence. The proposed algorithm consists of three stages: consensus embedding, cluster-center selection, and clustering assignment.

In the consensus embedding stage, constructing the Laplacian matrix from the $n \times n$ adjacency matrix requires $O(\mathrm{nnz}(A))$ time, where $\mathrm{nnz}(A)$ is the number of nonzero entries in A. Computing eigenpairs of a dense $n \times n$ matrix using the classical QR algorithm requires $O(n^3)$ time,[45] but this cost can be reduced to $O(\mathrm{nnz}(A) \cdot t + nt^2)$ when using the Krylov–Schur algorithm.[46] When $|E| = \Theta(n)$ and $t \ll n$, the time complexity reduces to $O(n)$, which is a common case in network analysis. In the attribute-embedding stage, normalizing the $n \times d$ attribute matrix requires $O(nd)$, computing the cosine similarity matrix requires $O(n^2 d)$, and the subsequent eigen decomposition requires $O(n^2 t)$ when using the Krylov–Schur algorithm. The CCA-based fusion step costs $O(nt^2 + t^3)$, where the first term comes from computing covariance matrices and the second from solving a $2t \times 2t$ eigenproblem.

In the cluster center selection stage, computing all pairwise distances and local densities requires $O(n^2)$, and selecting cluster centers adds $O(n \log C)$, resulting in an overall complexity of $O(n^2)$. In the clustering assignment stage, computing distances between n nodes and C cluster centers requires $O(nC)$, and iterative membership updates over T iterations contribute $O(TnC)$. Therefore, this stage incurs $O(TnC)$ in total.

Combining all stages, the total worst-case time complexity is $O(n^3)$ when the adjacency matrix is dense and the classical QR algorithm is used. However, in the common case where the adjacency matrix is sparse and the Krylov–Schur algorithm is applied, the overall complexity can be significantly reduced to $O(n^2(d+t) + TnC)$, making the algorithm scalable and efficient for network analysis.

6. Experiment

In this section, the performance of the DPCNCM model is evaluated through a series of experiments. The experiment aims to assess the effectiveness of the DPCNCM model by comparing it with other algorithms on both synthetic and real-world network datasets. An overview of this section is as follows. In Subsection 6.1, the datasets used in the experiments, including both synthetic and real-world networks, are presented. In Subsection 6.2, several widely used comparison methods are introduced. This section also discusses the metrics employed to assess the detected overlapping communities.

The experimental results on different datasets are analyzed in Subsections 6.3 and 6.4 where the overlapping nodes detected by the DPCNCM algorithm are showcased by network structure diagram in Subsection 6.4. Some crucial parameters are adjusted to analyze the sensitivity of the DPCNCM model in Subsection 6.5. For the detailed code and data, please refer to authors' website.

6.1. Datasets description

To evaluate the performance of the DPCNCM algorithm, we use two types of datasets: a synthetic dataset and a real-world dataset. The synthetic dataset consists of artificially generated networks, which allow us to establish a ground truth for comparing the communities generated synthetically with those detected by the algorithm.

Synthetic datasets We construct synthetic networks with overlapping communities using the LFR benchmark model, which has been repeatedly validated as an effective method for generating social networks.[47] The dataset is defined by several parameters: N represents the total number of nodes. The mixing parameter μ determines the proportion of edges that connect nodes from different communities. The average node degree k, as well as the minimum and maximum community sizes, can be manually adjusted to control the range of community memberships. Additionally, the datasets include a specified number of overlapping nodes O_n, with each of these nodes belonging to two communities. The generated output consists of a graph G and a collection of community sets, where each set contains the nodes forming a specific community. The two networks we generated are shown in Table 1.

Real-world datasets In addition to the artificial networks, we evaluated our method on five classic real-world networks widely studied in network science. Their node and edge counts, together with brief descriptions of their node attributes, are listed in Table 2.

Table 1. Artificial synthetic network datasets.

Name	N	μ	O_n	k
NetworkA	5000	0.1–0.6	50	18
NetworkB	2000–10000	0.2	20–100	18

Table 2. Real-world network datasets.

Network	Nodes	Edges	Node attributes
American Football Games[48]	115	613	conference (12 classes)
Political Books[49]	105	441	political stance (3 classes)
Cornell (WebKB)[50]	183	298	web-page category (5 classes)
Texas (WebKB)[50]	187	325	web-page category (5 classes)
Wisconsin (WebKB)[50]	251	515	web-page category (5 classes)

6.2. Baseline methods and evaluation metrics

Several widely used methods are chosen as baselines in our experiments: FCM,[40] DP clustering,[20] COPRA,[34] the CPM[36] and DEMON.[35] The first two, FCM and DP, supply the core techniques embedded in DPCNCM, whereas the latter three are widely regarded as classic algorithms for detecting overlapping communities in networks.

We utilize two types of evaluation metrics to assess the quality of communities detected by different methods: information recovery-based metrics when ground-truth communities are available, and quality-based metrics when ground truth is absent.

When ground truth is available, we adopt two widely used metrics for overlapping community evaluation:

Overlapping normalized mutual information (ONMI)[51] An extension of normalized mutual information (NMI)[52] that accounts for nodes belonging to multiple communities.

Omega index[53] An extension of the adjusted Rand index (ARI)[54,55] designed for overlapping community structures.

In the absence of ground truth, we use a quality-based metric for overlapping community evaluation:

Extended modularity (EQ)[56] An adaptation of Newman's modularity[57] to overlapping communities by weighting node contributions according to their memberships.

6.3. Analysis of synthetic network results

In this section, we assess the performance of DPCNCM and the comparison algorithms on synthetic networks. Since ground-truth communities are available in synthetic networks, we use ONMI and Omega as evaluation metrics. To examine the performance of the algorithms under different levels of network complexity, we vary the mixing parameter μ and the number of nodes N. For DPCNCM, we set the consensus matrix dimension to $t = 5$, with parameters $\delta = 0.3$, $w_1 = 0.8$, $w_2 = 0.1$, and $w_3 = 0.1$, where δ controls the penalty for assigning nodes to the unassigned set, and w_1, w_2, w_3 are the weight coefficients for the determinate, boundary, and unassigned memberships, respectively. For algorithms with inherent randomness, we report the average results over five runs.

As shown in Fig. 2, DPCNCM outperforms all other algorithms, except at $\mu = 0.25$, where COPRA surpasses DPCNCM in ONMI by 0.026. When $\mu \leq 0.2$, both DPCNCM and COPRA achieve ONMI scores above 0.9, indicating that these methods can accurately identify well-defined commu-

nity structures. CPM performs the worst among all algorithms, with ONMI consistently below 0.3. As μ increases, the ONMI scores of all algorithms decline significantly due to the increasing number of inter-community edges, which obscure community boundaries and complicate detection. When $\mu > 0.5$, the ONMI values of all algorithms drop below 0.4. However, DPCNCM maintains the best performance due to its slower decline compared to COPRA. When evaluated using Omega, DPCNCM exhibits a similar performance decline in Omega as μ increases. DPCNCM achieves the best performance for $\mu < 0.6$, while DP surpasses DPCNCM by 0.031 in ONMI. Notably, DP experiences a slower decline in performance than other methods, resulting in higher Omega scores than DPCNCM for $\mu = 0.6$. COPRA achieves the second-best performance at $\mu = 0.1$ and $\mu = 0.2$, but its Omega scores approach zero when $0.3 \leq \mu \leq 0.5$.

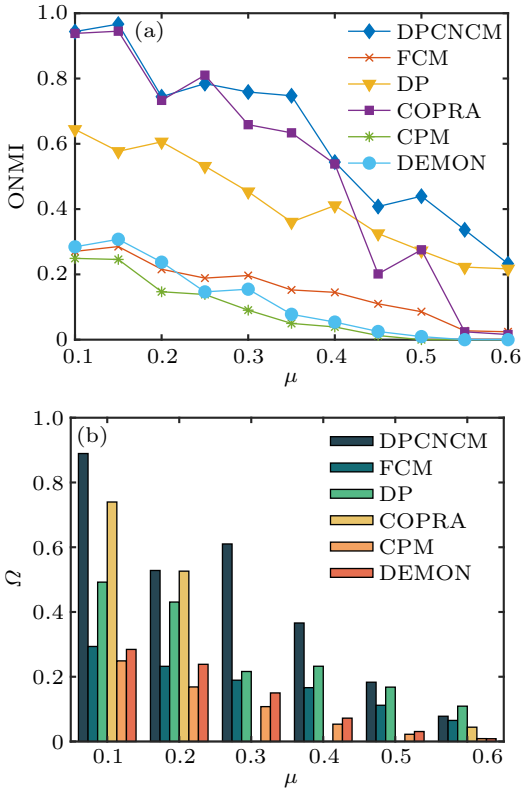


Fig. 2. Performance comparison of different algorithms on synthetic networks with varying mixing parameter $\mu \in [0.1, 0.6]$. (a) Evaluation using the ONMI metric, where higher values indicate better detection accuracy of overlapping community structures. (b) Evaluation using the Omega index metric, which measures the similarity between detected and ground-truth communities.

Next, we analyze the impact of different node numbers on all algorithms. As shown in Fig. 3, although the ONMI scores of DPCNCM exhibit a slight decrease at $N = 10000$, its overall performance remains stable, consistently ranking first among all algorithms. The stable performance across networks of different sizes suggests that DPCNCM is well-suited for large-scale datasets. COPRA performs as the second-best method for $2000 \leq N \leq 8000$, but its ONMI score drops to only 0.2264 in the network with 10000 nodes. CPM does not show a clear performance decline as the number of nodes increases; instead, its ONMI scores improve in networks with 2000 to 6000 nodes, although its performance remains inferior to DPCNCM. DEMON experiences a significant performance drop as the number of nodes increases, with its ONMI value approaching zero when $N \geq 8000$, indicating its unsuitability for large-scale tasks. When evaluated using the Omega metric, DPCNCM consistently outperforms other algorithms across all network sizes. It maintains a stable Omega score between 0.84 and 0.87 for $N < 10000$, though it drops to 0.5193 at $N = 10000$. The performance of DP and FCM does not exhibit a significant decline as the number of nodes increases, despite their dependence on initial cluster centers. In contrast, COPRA, CPM and DEMON show a clear performance drop as the network size grows.

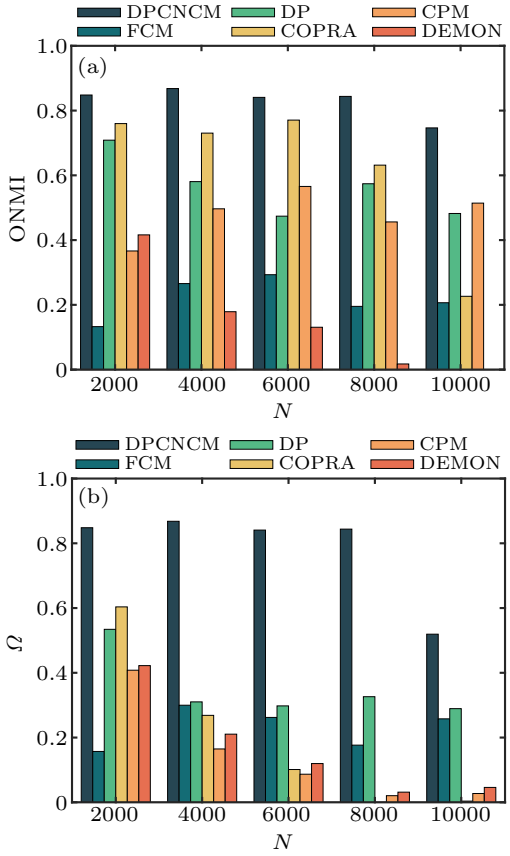


Fig. 3. Performance comparison of different algorithms on synthetic networks with varying node numbers $n \in [2000, 10000]$. (a) Evaluation using the ONMI metric, where higher values indicate better detection accuracy of overlapping community structures. (b) Evaluation using the Omega index metric, which measures the similarity between detected and ground-truth communities.

6.4. Analysis of real-world network results

In this section, we evaluate the performance of DPCNCM and comparison algorithms on real-world networks, including Football, Palbooks, Cornell, Texas, and Wisconsin. The parameter settings of DPCNCM in real-world networks are $\delta = 1$, $\omega_1 = 0.33$, $\omega_2 = 0.33$ and $w_3 = 0.33$. The number

of communities in each network is identified by DPCNCM, and we visualize the optimal number of clusters found by the piecewise least squares method for four of these networks, as shown in Fig. 4. The specific number of communities detected for each real-world network is listed in Table 3.

Table 3. Number of communities detected by DPCNCM in real-world networks.

	Football	Polbooks	Cornell	Texas	Wisconsin
Number of communities	11	2	9	8	7

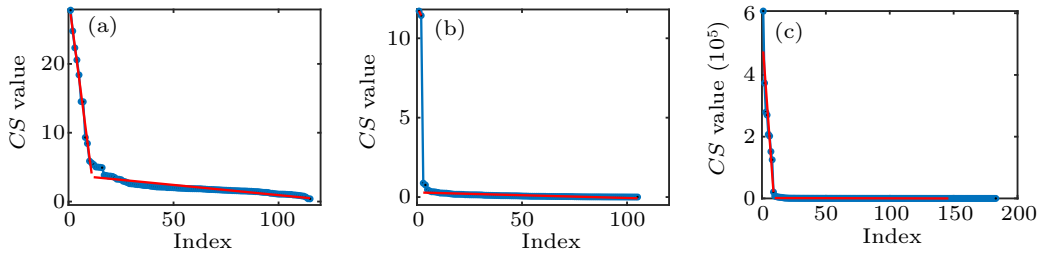


Fig. 4. DPCNCM community detection in real-world networks. The breakpoints of two red fitting lines indicate the optimal numbers. (a) 11 communities in the football network; (b) 2 communities in the political books network; (c) 9 communities in the Cornell network.

Table 4. Comparison of ONMI and EQ scores for different algorithms on real-world networks.

	Network	DPCNCM	FCM	DP	COPRA	CPM	DEMON
ONMI	Football	**0.8302**	0.5621	0.7103	0.5791	0.4313	0.3151
	Polbooks	**0.5325**	0.3363	0.3388	0.4005	0.2304	0.3204
EQ	Cornell	**0.4796**	0.4529	0.2176	0.4146	0.2422	0.1959
	Texas	**0.3794**	0.3415	0.0999	0.3210	0.2033	0.0599
	Wisconsin	0.4413	0.3902	0.0179	**0.5148**	0.2298	0.0983

For the Football and Polbooks networks, each node has a given value representing either its conference affiliation or political stance, respectively. Therefore, ONMI is used to assess how well the detected communities align with these ground-truth categories. For the other networks, we evaluate the quality of the detected community partitions using modularity, with all results presented in Table 4. As shown, DPCNCM achieves the highest ONMI score among the networks with given labels, indicating its effectiveness in discovering communities that correspond well to known categories. For the three WebKB networks without predefined labels, DPCNCM obtains the highest EQ scores on Cornell and Texas, but shows no clear advantage overall, and is even outperformed by COPRA on Wisconsin by 0.0735. Given the sparsity of the WebKB networks, this lack of advantage under the EQ metric suggests a limitation of DPCNCM in capturing structural patterns when topological signals are weak. In addition to its clustering performance, DPCNCM converges within 44, 30, 43, 77, and 98 iterations on the Football, Polbooks, Cornell, Texas, and Wisconsin datasets, respectively, highlighting its efficiency and practicality for overlapping community detection on attributed graphs.

To further illustrate DPCNCM's interpretability in identifying overlapping communities, we visualize the detected community structures in the Football and Palbooks networks, as shown in Fig. 5. In these figures, numbers on the nodes represent their IDs, and nodes that belong to overlapping communities are shown in two colors, highlighting their membership in multiple communities. For example, in the Football network, nodes 36, 80, and 82 are identified as members of overlapping communities. In the Palbooks network, DPCNCM uncovers two distinct communities along with an overlapping region containing 12 nodes, which align with the ground-truth categories: "liberal", "neutral", and "conservative". The interpretability of DPCNCM is reflected in the structured and semantically patterns it reveals, which provide intuitive insights into the network's organization and facilitate expert understanding.

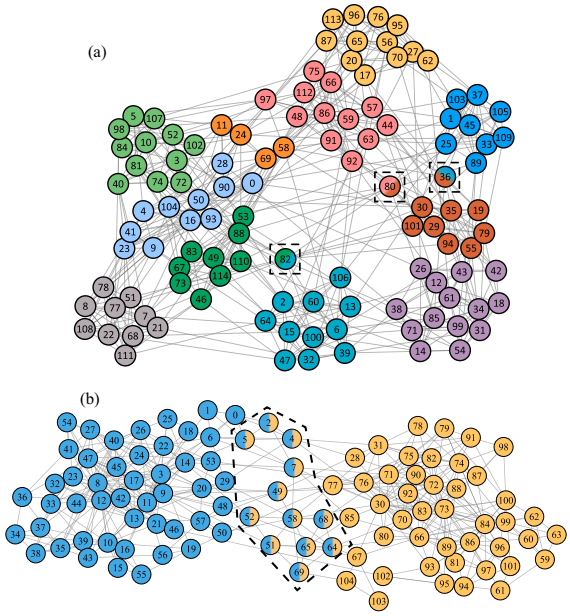


Fig. 5. The overlapping community structure detected by the DPCNCM algorithm, where nodes in different communities are colored distinctly, and nodes belonging to overlapping communities are represented with dual colors. (a) Visualization of the American football games network. (b) Visualization of the political books network.

6.5. Analysis of parameter influence

In this section, we analyze the influence of key parameters in DPCNCM, including t, w_1, w_2, w_3, and δ. The impact of dimension t of consensus embedding matrix is evaluated on synthetic networks using the ONMI metric, while the effects of w_1, w_2, w_3, and δ are examined on the real-world network.

We test different values of the consensus matrix dimension t ranging from 3 to 7 on NetworkA, with the corresponding ONMI results presented in Fig. 6. The results indicate that when t is small, DPCNCM exhibits weaker performance in detecting distinct community structures ($\mu < 0.4$). For example, in a network with $\mu = 0.2$, DPCNCM achieves an ONMI score of 0.7618 when $t = 7$, whereas the ONMI drops to 0.6514 when $t = 3$. However, as the network structure becomes more complex ($\mu > 0.4$), the performance of DPCNCM with a larger t deteriorates rapidly, leading to lower ONMI scores compared to a smaller t. For instance, in a network with $\mu = 0.6$, DPCNCM with $t = 7$ achieves an ONMI score of only 0.0954, while setting $t = 4$ results in a significantly higher ONMI of 0.2405. Based on these observations, we set $t = 5$ as it provides a balanced performance across both distinct and complex networks, ensuring stable results.

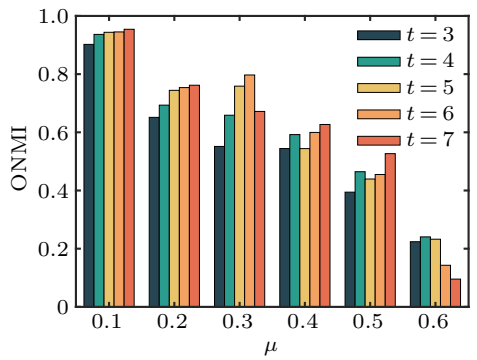


Fig. 6. The influence of parameter t on ONMI in synthetic networks with $\mu \in [0.1, 0.6]$.

Additionally, we analyze the effect of different weight parameters on the membership matrix in the real-world Cornell network. The number of nodes assigned to distinct communities, overlapping communities, and the unassigned set under different weight settings is summarized in Table 5. In the table, N_T represents the number of nodes assigned to distinct communities, N_I denotes the number of nodes belonging to overlapping communities, and N_F indicates the number of unassigned nodes that are not assigned to any distinct or overlapping community. It can be observed that as the value of δ decreases, the cost of placing a node in the unassigned set becomes lower, and the model tends to label nodes that are distant from cluster centers as unassigned nodes rather than forcing them into a determinate or overlapping community. For instance, when $\delta = 0.1$, a large number of nodes are identified as unassigned nodes. Moreover, the values of N_T, N_I, and N_F also vary inversely with the weight parameters ω_1, ω_2, and ω_3. The balanced weight setting ($\omega_1 = \omega_2 = \omega_3 = 0.33$) demonstrates strong robustness, as it yields relatively stable partitioning results regardless of the δ value. Due to the low proportion of overlapping nodes in the synthetic networks, we selected parameter settings that emphasize structural information while retaining limited sensitivity to overlapping memberships. This configuration is also consistent with the settings adopted in previous work.[19] For real-world networks, where ground-truth labels are unavailable, we adopted a balanced parameter setting to ensure robustness and avoid bias in the community detection process.

Table 5. Summary of node assignments under different weight settings.

δ	ω_1	ω_2	ω_3	N_T	N_I	N_F
0.3	0.8	0.1	0.1	173	10	0
	0.6	0.2	0.2	176	7	0
	0.33	0.33	0.33	182	1	0
0.2	0.8	0.1	0.1	158	7	18
	0.6	0.2	0.2	175	7	1
	0.33	0.33	0.33	182	1	0
0.1	0.8	0.1	0.1	50	0	133
	0.6	0.2	0.2	145	0	38
	0.33	0.33	0.33	182	0	1

7. Conclusion

To explore the overlapping community detection on attributed graphs, in this article, we propose an improved community detection method based on NCM theory. This method utilizes CCA to integrate node attribute information and topological relationship information, embedding them into Euclidean space. The number of clusters and initial centroids are then determined using the density peaks method. Building upon this, a FCM clustering algorithm which incorporating NCM theory is introduced, enabling effective community detection in attributed graphs. Results on both synthetic and real-world datasets demonstrate that the proposed algorithm outperforms baseline methods in detecting overlapping communities in attributed networks, although this advantage becomes less pronounced in sparse real-world networks, suggesting a direction for future improvement. Moreover, the DPCNCM algorithm exhibits robustness across networks with varying mixing parameters and node counts. Overall, the proposed method provides a solid foundation for advancing overlapping community detection in complex attributed networks. In future work, incorporating dynamic community detection strategies and a distributed parallel computing framework could further enhance its scalability and adaptability.

Program availability

The source code is openly available at
https://github.com/Yuhan-Jia/Code-for-DPCNCM

Acknowledgement

This work is supported by the Natural Science Foundation of China (Grant No. 72571150).

References

[1] Su X, Xue S, Liu F, Wu J, Yang J, Zhou C, Hu W, Paris C, Nepal S, Jin D, Sheng Q Z and Yu P S 2024 *IEEE Trans. Neural Netw. Learn. Syst.* **35** 4682
[2] Ma J L, Li M and Li H J 2024 *IEEE Trans. Netw. Sci. Eng.* **11** 14
[3] Cao J, Xu W, Jin D, Zhang X, Liu L and Miller A 2025 *Inf. Sci.* **686** 121349
[4] Gmati H, Mouakher A, Gonzalez-Pardo A and Camacho D 2024 *J. Ambient Intell. Humaniz. Comput.* **15** 1779
[5] Ji F, Wu J, Chiclana F, Wang S, Fujita H and Herrera-Viedma E 2023 *IEEE Trans. Fuzzy Syst.* **31** 3025
[6] Costa L F, Oliveira Jr O N, Travieso G, Rodrigues F A, Villas Boas P R, Antiqueira L, Viana M P and Correa Rocha L E 2011 *Adv. Phys.* **60** 329
[7] Li H J, Lou F H, Wang Q Q and Li G J 2025 *Chaos Soliton Fract.* **199** 116654
[8] Bothorel C, Cruz J D, Magnani M and Micenkova B 2015 *Netw. Sci.* **3** 408
[9] Lu H, Hong H and Geng X 2024 *Neural Process. Lett.* **56** 130.
[10] Berahmand K, Nasiri E and Li Y 2021 *Comput. Biol. Med.* **138** 104933
[11] Grassi R, Bartesaghi P, Benati S and Clemente G P 2021 *Netw. Spat. Econ.* **21** 707
[12] Zhang Y, Pan R, Wang H and Su H 2023 *Stat* **12** 507
[13] Fortunato S 2010 *Phys. Rep.* **486** 75
[14] Grover A and Leskovec J *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, August 13–17, 2016 San Francisco, United States, p. 855
[15] Berahmand K, Mohammadi M, Saberi-Movahed F, Li Y and Xu Y 2022 *IEEE Trans. Netw. Sci. Eng.* **10** 372
[16] de Silva A, Chen G, Ma H, Nekooei S M, Zuo X 2025 arXiv:2505.10197 [cs.SI]
[17] Zhang A, Wang X, Zhao Y 2024 arXiv:2411.01947 [cs.SI]
[18] Zeng R, Liu Z, Guo K 2024 *Computer Supported Cooperative Work and Social Computing (ChineseCSCW 2023)* 137.
[19] Guo Y and Sengur A 2015 *Pattern Recognit.* **48** 2710
[20] Rodriguez A and Laio A 2014 *Sci.* **344** 1492
[21] Costa A R 2021 arXiv:2111.15623 [cs.LG]
[22] Xie Y, Wang X, Jiang D, Xu R 2019 *Inf. Sci.* **501** 1
[23] Al-Garadi M A, Varathan K, Dey S and Sarker S S 2022 *J. Eng.* **2022** 1
[24] Liu R, Wang H and Yu X 2018 *Inf. Sci.* **450** 200
[25] Du M, Ding S and Xue Y 2018 *Int. J. Mach. Learn. Cybern.* **9** 1131
[26] Li J, Dani H, Hu X, Tang J, Chang Y and Liu H 2017 *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, November 6–10, 2017 Singapore, Singapore, p. 387
[27] Pearson K 1905 *Nature* **72** 342
[28] Zhou Y, Cheng H, Yu J X 2009 *Proc. VLDB Endow.* **2** 718
[29] Gao H and Huang H 2018 *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, July 13–19, 2018 Stockholm, Sweden, p. 3364
[30] Tang J, Qu M, Wang M, Zhang M, Yan J and Mei 2015 *Proceedings of the 24th international conference on world wide web*, May 18–22, 2015 Florence, Italy, p. 1067
[31] Li H J, Lou F H, Wang Q Q, Li G J and Perc M 2025 *IEEE Trans. Netw. Sci. Eng.* **14** 1
[32] Chen J, Ma T and Xiao C Fastgcn 2018 arXiv: 1801.10247 [cs.LG]
[33] Belkin M and Niyogi P 2003 *Neural. Comput.* **15** 1373
[34] Gregory S 2010 *New J. Phys.* **12** 103018
[35] Coscia M, Rossetti G, Giannotti F and Pedreschi D 2012 *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, August 12–16, 2012 Beijing, China, p. 615
[36] Palla G, Derényi I, Farkas I and Vicsek, T 2005 *Nature* **435** 814
[37] Li H J, Song S P, Tan W Z, Huang Z C, Li X Y, Xu W Z and Cao J 2022 *Neurocomputing* **512** 482
[38] Li H J, Zhang J, Liu Z P, Chen L and Zhang X S 2012 *Eur. Phys. J. B* **85** 1
[39] Smarandache F 2001 arXiv: math/0101228 [math.GM]
[40] Bezdek J C, Ehrlich R and Full W 1984 *Comput. Geosci.* **10** 191
[41] Ma P J, Ren X Z, Zhu J F and Jiang Y J 2024 *Chin. Phys. B* **33** 088903
[42] Ai D, Liu X L, Kang W Z, Li L N, Lu S Q and Liu Y 2023 *Chin. Phys. B* **32** 118902
[43] Su Z, Chen L, Ai J, Zheng Y Y and Bie N 2024 *Chin. Phys. B* **33** 058901
[44] Wang J, Zhang Y and Lan X 2016 *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, Oct 14–17, 2016 Chengdu, China, p. 13
[45] Parlett B N 2002 *Comput. Sci. Eng.* **2** 38
[46] Stewart G W 2002 *SIAM J. Matrix Anal. Appl.* **23** 601
[47] Lancichinetti A, Fortunato S and Radicchi F 2008 *Phys. Rev. E:Stat., Nonlinear, Soft Matter Phys.* **78** 046110
[48] Girvan M and Newman M E 2002 *Proc. Natl. Acad. Sci. USA* **99** 7821
[49] Krebs V http://www.orgnet.com/ [2025-3-31]
[50] Craven M, DiPasquo D, Freitag D, McCallum A, Mitchell T, Nigam K and Slattery S 1998 *web. AAAI/IAAI* **3** 2
[51] Lancichinetti A, Fortunato S and Kertész J 2009 *New J. Phys.* **11** 033015
[52] Strehl A and Ghosh J 2003 *J. Mach. Learn. Res.* **3** 3
[53] Collins L M and Dent C W 1988 *Multivar. Behav. Res.* **23** 231
[54] Rand W M 1971 *J. Am. Stat. Assoc.* **66** 846
[55] Hubert L and Arabie P 1985 *J. Classif.* **2** 193
[56] Shen H, Cheng X, Cai K and Hu M B 2009 *Phys. A* **388** 1706
[57] Newman M E and Girvan M 2004 *Phys. Rev. E* **69** 026113